

Abstract Number: 385
Last Modified: January 10 2009

Submitted By: Bennett Landman

A Rapid Prototyping Environment for NeuroImaging in Java

B A Landman¹, B Lucas^{2,3}, J A Bogovic², A Carass², J L Prince²

¹Department of Biomedical Engineering, Johns Hopkins University, Baltimore, MD, United States/²Department of Electrical and Computer Engineering, Johns Hopkins University, Baltimore, MD, United States/³Laboratory of Personality and Cognition, National Institute on Aging, National Institutes of Health, Baltimore, MD, United States

Introduction: Non-invasive neuroimaging techniques have enabled extraordinarily sensitive and specific study of the structure, functional response and connectivity of in vivo biological mechanisms. With these advanced methods comes a heavy reliance on computer-based processing, analysis and interpretation, which is non-trivial. While the neuroimaging community has produced many excellent academic and commercial tool packages, new tools are often required to interpret new modalities and paradigms. Developing custom tools and ensuring interoperability with existing tools is a significant hurdle. To address these limitations, we present a new framework for algorithm development which implicitly ensures interoperability, generates graphical user interfaces, provides advanced batch processing tools, and, most importantly, requires minimal additional programming or computational overhead. The system enables rapid prototyping of neuroimaging tools.

Methods: We developed a rapid prototyping extension of MIPAV (Medical Image Processing, Analysis and Visualization, <http://mipav.cit.nih.gov/>). With the new system, a programmer need only define the input and output requirements of her program and implement the core features (i.e., the algorithm). File reading, image format conversion, argument type checking, and scheduling are handled by the infrastructure rather than the programmer. All data are available at the code level as both objects (which can be manipulated with sophisticated operations – e.g., registration, segmentation, algebra, etc.) and as native Java arrays (e.g., 8/16/32 bit integers, 32/64 bit floating point, complex numbers, color codes, etc.). This dual perspective enables the programmer to choose the programming level at which she can be most productive and most easily implement the desired algorithm. Authors may include description information about tools usage, relevant citations, authors, affiliations, and version history which is available to the end-user with all usage types. Since platform uses Java, the classes may be interactively accessed and tested through the Matlab (Mathworks, Natick, MA) interpreter, which is an extremely popular and flexible prototyping language.

Results: Algorithms implemented within this framework can be accessed on any Java-enabled platform in three ways: (1) as plugins to the MIPAV program (Figure 1), (2) through a graphical pipeline environment (Figure 2 (Lucas, B, 2008)), and (3) as command line modules, which may be used in any scripting or command line environment. Workflows generated in the pipeline environment may be used for batch processing (Figure 3), in which dependencies are automatically resolved across multiple CPUs. Since MIPAV handles all format conversion, imaging data can easily be processed for FSL/FreeSurfer (NIFTI), SPM (NIFT, Analyze), and/or AFNI (BRICK) compatibility. MIPAV natively supports file formats for over 40 other imaging platforms as well.

Conclusions: Java-based rapid prototyping with this system is an efficient and practical method to evaluate new algorithms since the proposed system ensures that rapidly constructed prototypes are actually fully-functional processing modules with support for multiple GUI's, a broad range of file formats, and distributed computing features. Our group and collaborators have implemented over 100 tools for tasks including registration, segmentation, diffusion tensor imaging, cortical surface extraction and analysis, and sub-cortical processing. The system is released in open source under LGPL (Lesser GNU Public License) on NITRC (Neuroimaging Informatics Tools and Resources Clearinghouse : <http://www.nitrc.org/projects/maps4mipav/> [infrastructure], <http://www.nitrc.org/projects/jhumipavplugins/> [tools]).

References:

Lucas, B (2008), 'MAPS: A Free Medical Image Processing Pipeline', *Organization for Human Brain Mapping Conference*, vol. Melbourne, Australia, no. , pp. 512 W-PM.

